Algoritmos de Divisão e Conquista:

Método Mestre para resolver recorrências

- Além dos métodos de "Desenrolar" a recursão (recurrence trees) e "Chute" inicial (substitution method) existe um método alternativo para solucionar problemas genéricos de recorrência.
- O método mestre foi apresentado incialmente por Jon Bentley, Dorothea Blostein e James B. Saxe em 1980, como um método unificado para resolver recorrências.
- O Método Mestre provê uma "receita de bolo" para resolver recorrências de forma:

$$T(n) = aT(n/b) + f(n)$$

onde $a \geq 1$ e $b > 1$ são constantes e $f(n)$ é uma função assintoticamente positiva.

- Para aplicar o método mestre precisamos memorizar três casos e com eles seremos capazes de resolver muitos problemas de recorrência facilmente.

TEOREMA T.1 O método mestre

- **Permita que $a \geq 1$ e $b > 1$ sejam constantes, $f(n)$ é uma função e deixe que $T(n)$ seja definido por uma recorrência em inteiros não negativos.**

$$T(n) = aT(n/b) + f(n)$$

onde interpretamos que n/b pode significar $\lfloor n/b \rfloor$ ou $\lceil n/b \rceil$. Então $T(n)$ possuí os seguintes limites assintóticos:

- **Caso 1: Se $f(n) = O\left(n^{\log_b a-\epsilon}\right)$ para alguma constante $\epsilon > 0$, então $T(n) = \Theta\left(n^{log_b a}\right)$.**

- **Caso 2: Se $f(n) = \Theta\left(n^{\log_b a}\right)$, então $T(n) = \Theta\left(n^{log_b a} \log_2 n\right)$.**

- **Caso 3: Se $f(n) = \Omega\left(n^{\log_b a+\epsilon}\right)$ para alguma constante $\epsilon > 0$, e se $af(n/b) \leq cf(n)$ para alguma constante $c < 1$ e todos os n suficientemente grandes, então $T(n) = \Theta\big(f(n)\big)$.**

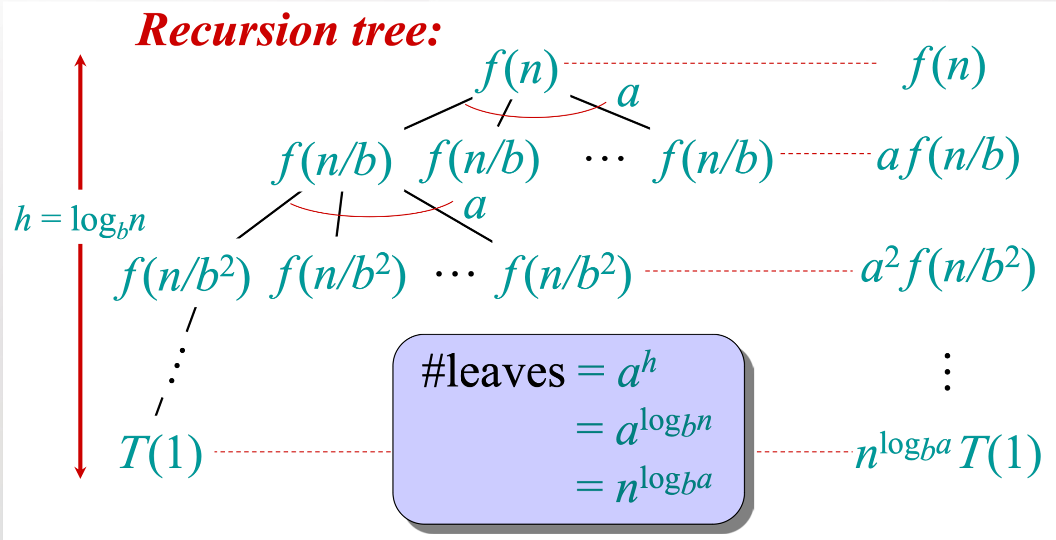
PUCRS ESCOLA POLITÉCNICA

Algoritmos de Divisão e Conquista:

Método Mestre para resolver recorrências

TEOREMA T.1 O método mestre

- ❖ **Caso 1: Se $f(n) = O\left(n^{\log_b a-\epsilon}\right)$ para alguma constante $\epsilon > 0$, então $T(n) = \Theta\left(n^{\log_b a}\right)$.**
- ❖ **Caso 2: Se $f(n) = \Theta\left(n^{\log_b a}\right)$, então $T(n) = \Theta\left(n^{\log_b a}\log_2 n\right)$.**
- ❖ **Caso 3: Se $f(n) = \Omega\left(n^{\log_b a+\epsilon}\right)$ para alguma constante $\epsilon > 0$, e se $af(n/b) \leq cf(n)$ para alguma constante $c < 1$ e todos os n suficientemente grandes, então $T(n) = \Theta\left(f(n)\right)$.**



- Dada a recorrência $T(n) = aT(n/b) + f(n)$, em cada um dos casos acima comparamos a função $f(n)$ com a função $n^{\log_b a}$ que nada mais é do que o custo todas de todas as folhas da nossa recorrência.
- O maior valor entre as duas funções determina a solução da recorrência.

PUCRS ESCOLA POLITÉCNICA

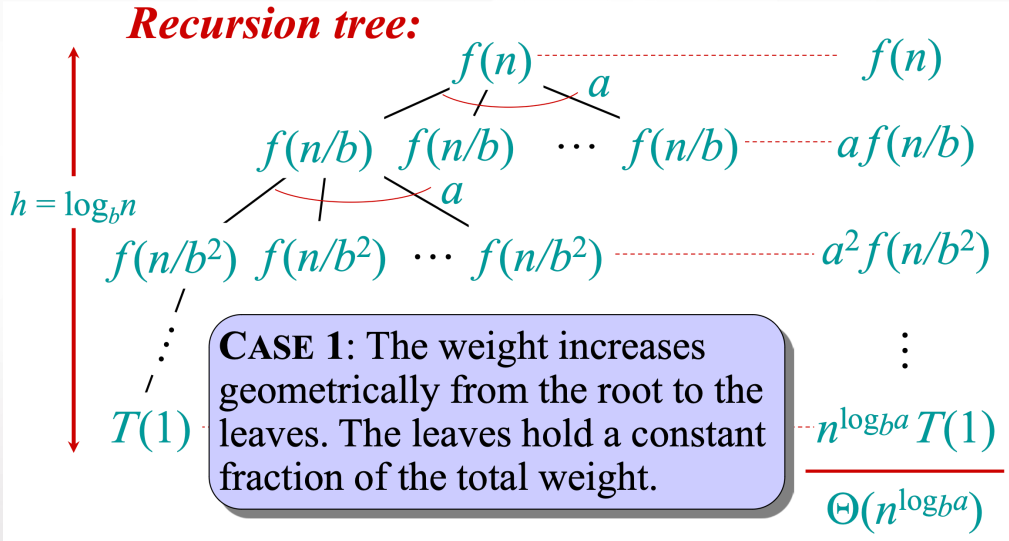
Algoritmos de Divisão e Conquista:

Método Mestre para resolver recorrências

TEOREMA T.1 O método mestre

- **Caso 1: Se $f(n) = O\left(n^{\log_b a-\epsilon}\right)$ para alguma constante $\epsilon > 0$, então $T(n) = \Theta\left(n^{\log_b a}\right)$.**
- **Caso 2: Se $f(n) = \Theta\left(n^{\log_b a}\right)$, então $T(n) = \Theta\left(n^{\log_b a} \log_2 n\right)$.**
- **Caso 3: Se $f(n) = \Omega\left(n^{\log_b a+\epsilon}\right)$ para alguma constante $\epsilon > 0$, e se $af(n/b) \leq cf(n)$ para alguma constante $c < 1$ e todos os n suficientemente grandes, então $T(n) = \Theta\left(f(n)\right)$.**

- No caso 1, $n^{\log_b a}$ é maior que $f(n)$, então o resultado é $T(n) = \Theta\left(n^{\log_b a}\right)$.



- $T(n) = aT(n/b) + f(n) = 4T(n/2) + n$

onde $a = 4, b = 2, f(n) = n$

- Então temos que $n^{\log_b a} = n^{\log_2 4} = n^2 = \Theta(n^2)$
- Como $f(n) = O\left(n^{\log_b a-\epsilon}\right)$, onde $\epsilon = 1$, pois $n^{\log_b a-\epsilon} = n^{\log_2 4-1} = n^{2-1} = n$.
- Portanto, $n^{\log_b a} > f(n)$, a solução é $T(n) = \Theta(n^2)$.

PUCRS ESCOLA POLITÉCNICA

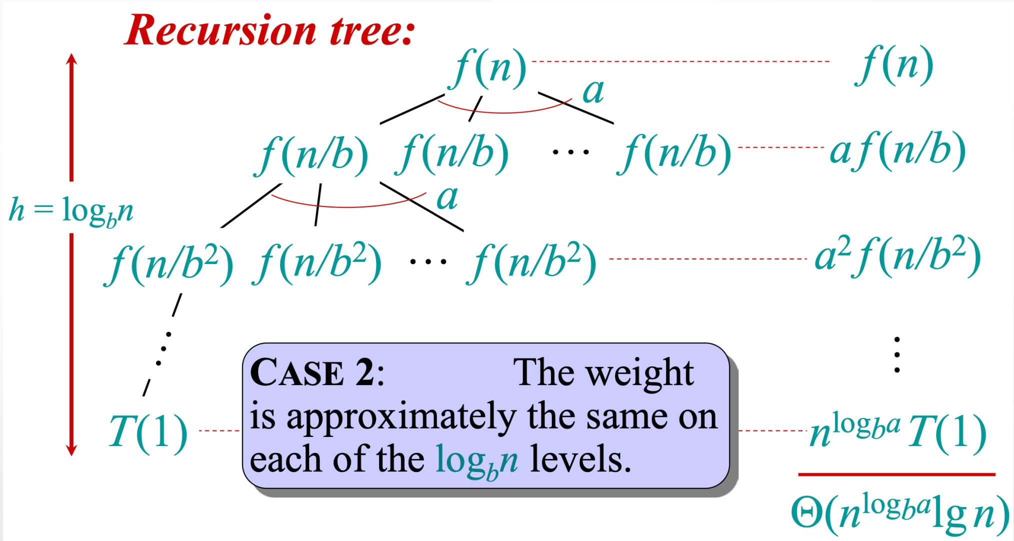
Algoritmos de Divisão e Conquista:

Método Mestre para resolver recorrências

TEOREMA T.1 O método mestre

- **Caso 1: Se $\boldsymbol{f(n) = O(n^{\log_b a-\epsilon})}$ para alguma constante $\boldsymbol{\epsilon > 0}$, então $\boldsymbol{T(n) = \Theta(n^{\log_b a})}$.**
- **Caso 2: Se $\boldsymbol{f(n) = \Theta(n^{\log_b a})}$, então $\boldsymbol{T(n) = \Theta(n^{\log_b a} \log_2 n)}$.**
- **Caso 3: Se $\boldsymbol{f(n) = \Omega(n^{\log_b a+\epsilon})}$ para alguma constante $\boldsymbol{\epsilon > 0}$, e se $\boldsymbol{af(n/b) \leq cf(n)}$ para alguma constante $\boldsymbol{c < 1}$ e todos os $\boldsymbol{n}$ suficientemente grandes, então $\boldsymbol{T(n) = \Theta(f(n))}$.**

- No caso 2, se as duas funções são do mesmo tamanho o resultado é $T(n) = \Theta(n^{\log_b a} \log_2 n) = \Theta(f(n) \log_2 n)$.



- $T(n) = aT(n/b) + f(n) = 4T(n/2) + n^2$

onde $a = 4, b = 2, f(n) = n^2$

- Então temos que $n^{\log_b a} = n^{\log_2 4} = n^2 = \Theta(n^2)$
- Como $f(n) = O(n^{\log_b a})$, pois $n^{\log_b a} = n^{\log_2 4} = n^2$.
- Portanto, $n^{\log_b a} = f(n)$, a solução é $T(n) = \Theta(n^2 \log_2 n) = \Theta(f(n) \log_2 n)$.

PUCRS ESCOLA POLITÉCNICA

Algoritmos de Divisão e Conquista:

Método Mestre para resolver recorrências

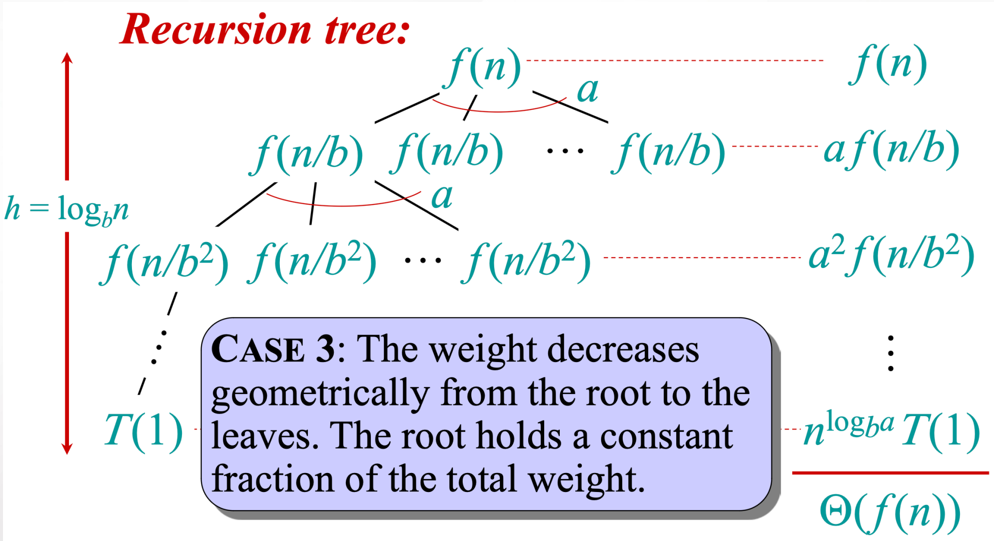
TEOREMA T.1 O método mestre

- **Caso 1: Se $f(n) = O(n^{\log_b a-\epsilon})$ para alguma constante $\epsilon > 0$, então $T(n) = \Theta(n^{\log_b a})$.**

- **Caso 2: Se $f(n) = \Theta(n^{\log_b a})$, então $T(n) = \Theta(n^{\log_b a}\log_2 n)$.**

- **Caso 3: Se $f(n) = \Omega(n^{\log_b a+\epsilon})$ para alguma constante $\epsilon > 0$, e se $af(n/b) \le cf(n)$ para alguma constante $c < 1$ e todos os n suficientemente grandes, então $T(n) = \Theta(f(n))$.**

- No caso 3, $n^{\log_b a}$ é menor que $f(n)$, então o resultado é $T(n) = \Theta(f(n))$.



- $T(n) = aT(n/b) + f(n) = 4T(n/2) + n^3$

onde $a = 4, b = 2, f(n) = n^3$

- Então temos que $n^{\log_b a} = n^{\log_2 4} = n^2 = \Theta(n^2)$

- Como $f(n) = \Omega(n^{\log_b a+\epsilon})$, onde $\epsilon = 1$, pois $n^{\log_b a+\epsilon} = n^{\log_2 4+1} = n^{2+1} = n^3$.

- $af(n/b) \le cf(n) \rightarrow 4(n/2)^3 \le cn^3 \rightarrow \frac{1}{2}n^3 \le cn^3$

onde $c = \frac{1}{2} < 1$.

- Portanto, $n^{\log_b a} < f(n)$, a solução é $T(n) = \Theta(n^3) = \Theta(f(n))$.